UNG Roland 20231933

BERNABEO Jeff 20232406

L3ASR

Rapport de projet

Thème 3 : attaque ICMP redirect

Projet administration système - ASR

Année universitaire 2025/2026

Table des matières

1. Outil Netcat

1.1. Présentation de l'outil Netcat

Netcat est un outil qui permet de créer des connexions réseau TCP et UDP, on l'utilise dans le projet pour établir une connexion entre le client et le serveur dans le but de transférer des données entre eux.

-l (mode listen) : netcat se met en écoute et attend qu'un client se connecte.

-p (port) : spécifie le port local d'écoute. Exemple : -p 9090 écoute sur le port TCP 9090.

1.2. Test de communication

Cas 1 : `root@debian:~# echo toto | nc 192.168.10.12 9090`

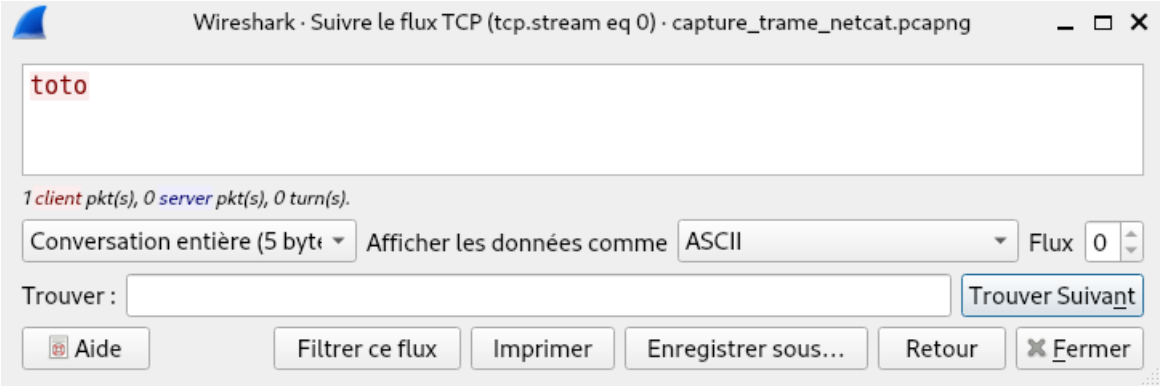
```
root@debian:~# nc -lp 9090
toto
```

nc -lp 9090 est d'abord tapé sur le serveur pour ouvrir la connexion, puis on envoie avec echo toto | nc 192.168.10.12 9090

Méthode 1 :

```
▸ Transmission Control Protocol, Src Port: 40152, Dst Port: 9090, Seq: 1, Ack: 1, Len: 5
▾ Data (5 bytes)
    Data: 746f746f0a
    [Length: 5]
```

Méthode 2 :



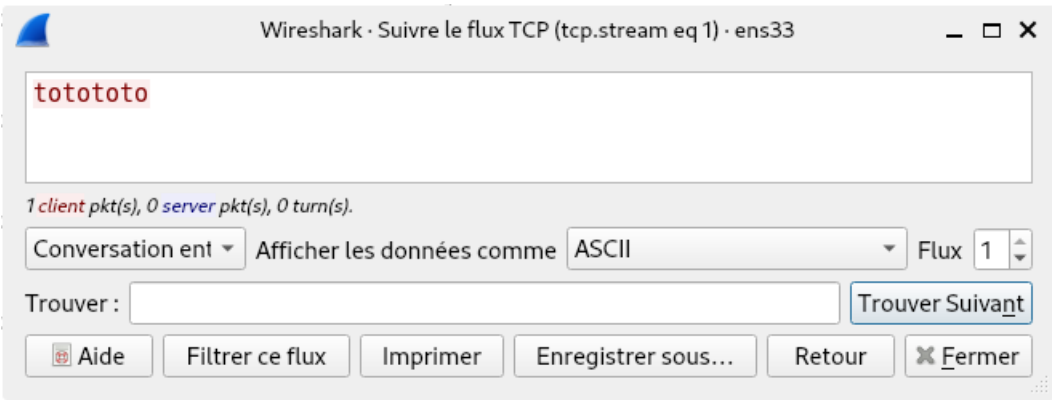
Le texte apparait en clair, pour vérifier cela, nous avons 2 méthodes, la première consiste à aller dans les détails du paquet TCP où l'on aperçoit le texte affiché en hexadécimal (en clair) et la deuxième méthode consiste à aller dans Analyser -> Suivre -> flux TCP à partir de là on aperçoit le texte « toto » en clair.

Cas 2 :

```
root@debian:~# nc 192.168.10.12 9090
totototo
```

Ici, on enlève echo toto et donc tous le texte qui est écrit sur le client est directement transmis au serveur et apparait dans son terminal.

```
▸ Transmission Control Protocol, Src Port: 43474, Dst Port: 9090, Seq: 1, Ack: 1, Len: 9
▾ Data (9 bytes)
    Data: 746f746f746f746f0a
    [Length: 9]
```



1.3. Transfert de fichier

Création du fichier « toto.txt » et ajout des lignes dans le client :

```
root@debian:~# echo "toto1 toto2" > toto.txt
root@debian:~# echo "toto3" >> toto.txt
root@debian:~# cat toto.txt
toto1 toto2
toto3
```

Envoi du contenu de « toto.txt » du client au serveur :

```
root@debian:~# nc 192.168.10.12 9090 < toto.txt
```

Création du fichier et reçu du contenu de « toto.txt » dans le fichier « received_toto.txt » côté serveur :

```
root@debian:~# nc -l -p 9090 > received_toto.txt
^C
root@debian:~# cat received_toto.txt
toto1 toto2
toto3
```

2. Outil Telnet

```
root@debian:~# telnet 192.168.10.7
```

```
Mot de passe : debian

Linux debian 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1
(2023-09-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Derni..re connexion : jeudi 20 novembre 2025 .. 10:30:28 CET de localhost
sur pts/1
.[?2004h.]0;debian@debian: ~..[01;32mdebian@debian.[00m:.[01;34m~.[00m$
```

On voit que la connexion n'est pas chiffrée. Le mot de passe est clairement visible, un attaquant ayant accès au trafic réseau pourrait intercepter la trame et lire en clair les identifiants, la connexion n'est pas sécurisée.
Le protocole Telnet est clairement obsolète aujourd'hui.

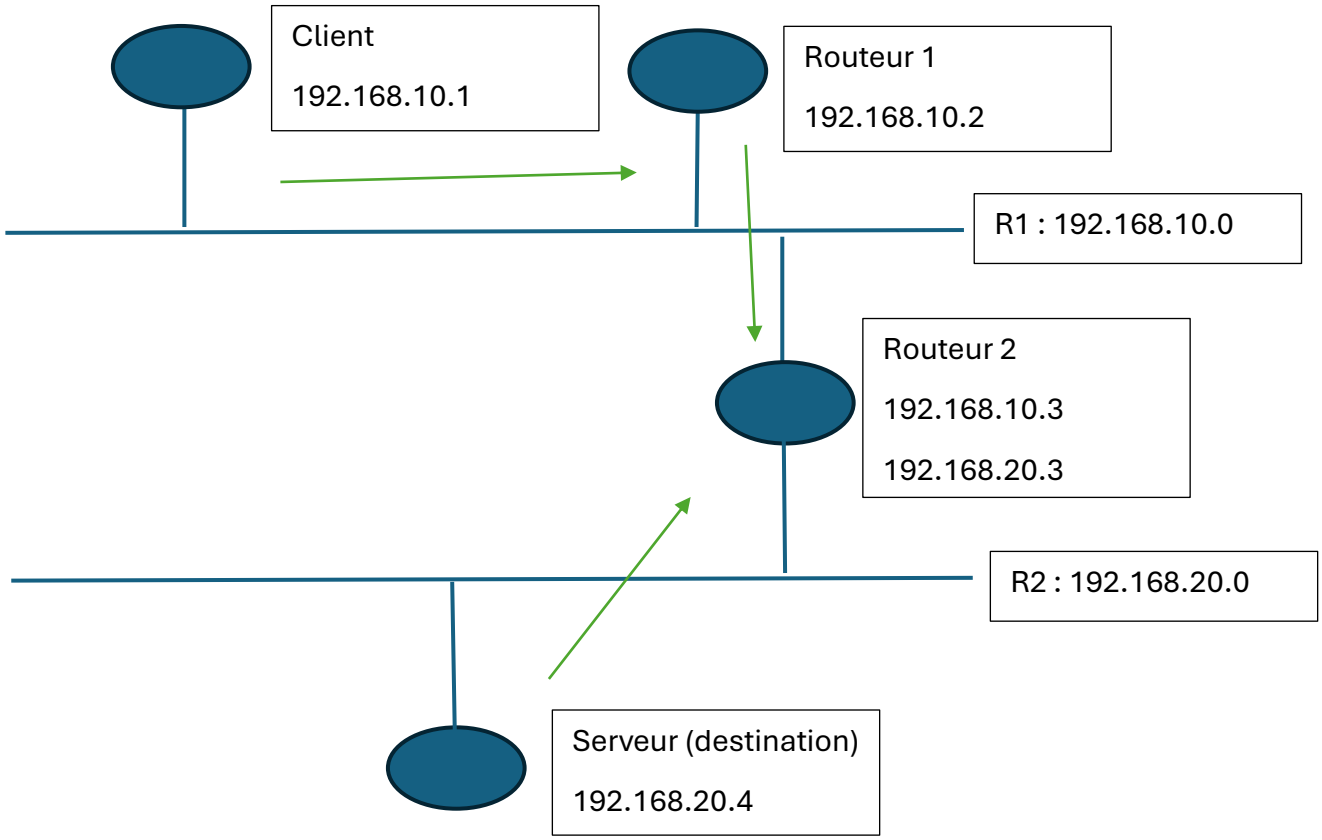
3. ICMP redirect

3.1. Introduction sur l'ICMP redirect

Lorsqu'une machine envoie un paquet vers une autre machine, le chemin utilisé peut ne pas être le plus optimale. Si c'est le cas, le routeur par lequel passe le paquet, envoie un message ICMP redirect à la machine source pour l'informer qu'il existe un chemin plus rapide via un autre routeur. Ainsi la machine met à jour sa table de routage pour envoyer directement son paquet vers ce routeur

3.2. Maquette pour illustrer le fonctionnement d'ICMP redirect

Dans la maquette, il y a deux réseaux. 3 machines sont sur le premier réseau et deux sur le second. Le routeur 2 possède deux interfaces, c'est le seul qui se trouve à la fois dans les deux réseaux. La passerelle par défaut du client est le routeur 1, puis celui du routeur 1 et du serveur, c'est le routeur 2



3.3. Fonctionnement de l'ICMP Redirect sur Wireshark

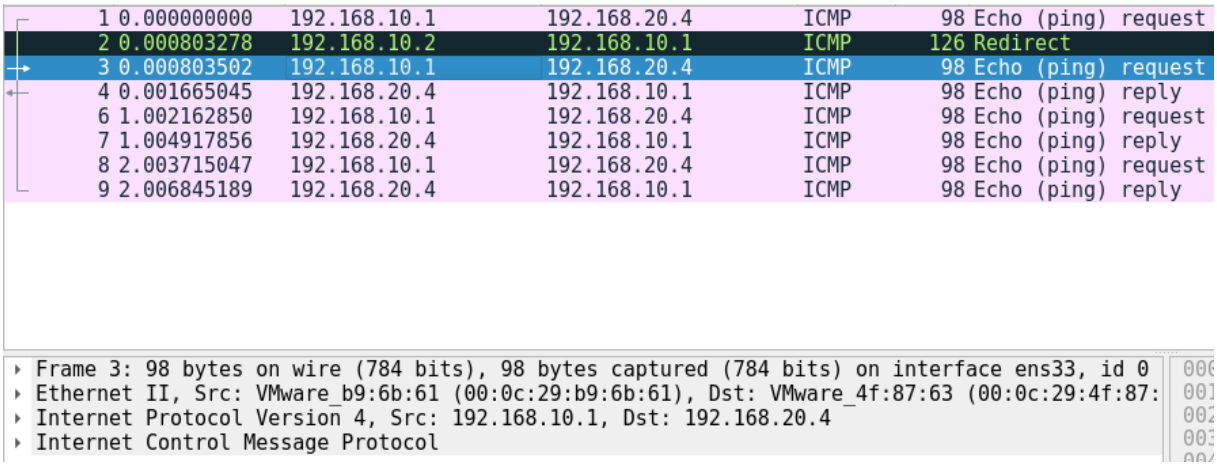
Captures de trames sur R1

1	0.000000000	192.168.10.1	192.168.20.4	ICMP	98 Echo (ping) request
2	0.000803278	192.168.10.2	192.168.10.1	ICMP	126 Redirect
3	0.000803502	192.168.10.1	192.168.20.4	ICMP	98 Echo (ping) request
4	0.001665045	192.168.20.4	192.168.10.1	ICMP	98 Echo (ping) reply
6	1.002162850	192.168.10.1	192.168.20.4	ICMP	98 Echo (ping) request
7	1.004917856	192.168.20.4	192.168.10.1	ICMP	98 Echo (ping) reply
8	2.003715047	192.168.10.1	192.168.20.4	ICMP	98 Echo (ping) request
9	2.006845189	192.168.20.4	192.168.10.1	ICMP	98 Echo (ping) reply

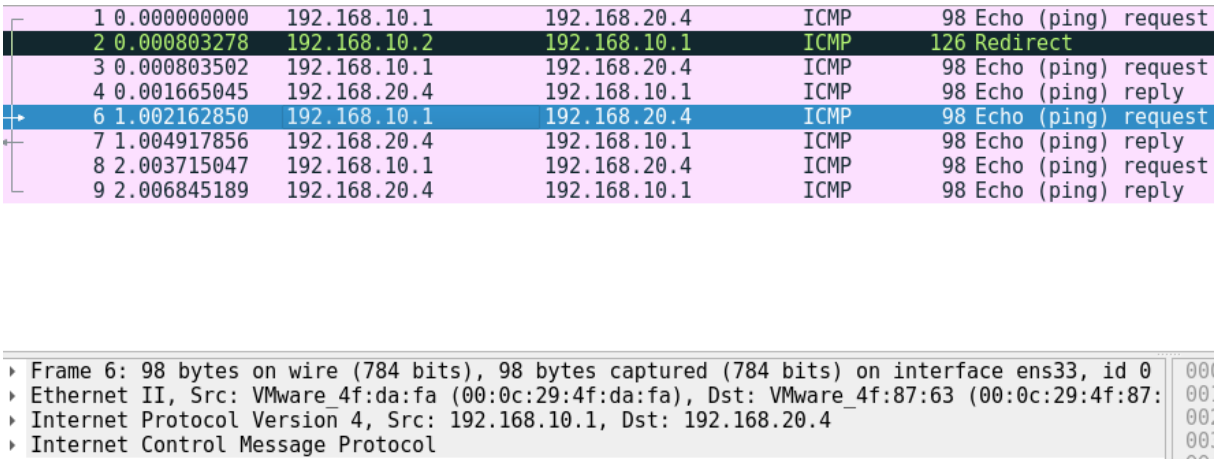
```
▸ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface ens33, id 0
▸ Ethernet II, Src: VMware_4f:da:fa (00:0c:29:4f:da:fa), Dst: VMware_b9:6b:61 (00:0c:29:b9:6b:
▸ Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.20.4
▸ Internet Control Message Protocol
```

Dans le premier échange de paquets, on voit que le trafic du client passe tout d'abord par le routeur 1. L'adresse MAC source correspond bien à celui du client (00 :0c :29 :4f :da :fa) et l'adresse MAC de destination à celui du routeur 1 (00 :0c :29 :b9 :6b :61).

Après avoir reçu ce premier paquet, le routeur 1 envoie un message ICMP Redirect au client pour dire qu'il existe une route plus optimale : celle en passant par le routeur 2 car elle est dans le même réseau que le serveur.

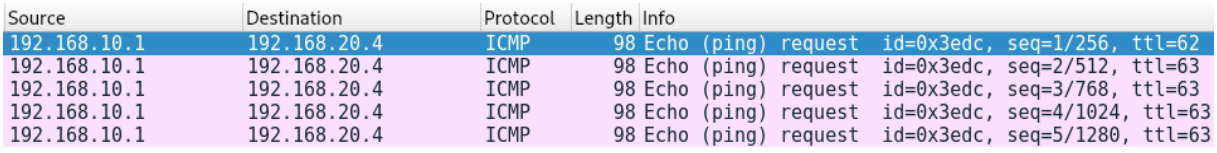


Même si le message ICMP Redirect est envoyé, le paquet qui était envoyé continue son chemin vers le serveur. On observe en effet dans la capture ci-dessus qu'il part du routeur 1 (00 :0c :29 :b9 :6b :61) et est ensuite transmis au routeur 2 (00 :0c :29 :4f :87 :63).



Mais pour les prochains paquets, grâce au message ICMP redirect, ils ne passeront plus par le routeur 1, mais directement par le routeur 2. Comme on le voit dans la capture, l'adresse MAC source est celui du client (00 :0c :29 :4f :da :fa), puis désormais l'adresse MAC de destination est cette fois ci celui du routeur 2 (00 :0c :29 :4f :87 :63). Le client a bien mis à jour sa table de routage après avoir reçu le message ICMP redirect.

Capture de trame sur R2



De plus, sur le réseau 192.168.20.0 on peut également vérifier que la table de routage du client est mise à jour puisqu'on voit que pour le premier échange de paquet, le TTL a diminué de 2 (c'est passé de 64 à 62), ce qui signifie que le paquet a traversé 2 machines (le routeur 1 et 2). En revanche pour les prochains envois de paquets, le TTL n'a diminué que de 1 (64 à 63), donc le paquet traverse bien qu'une seule machine, et il s'agit du routeur 2 comme montrer précédemment.

4. Attaque sur ICMP redirect

4.1. Présentation de la maquette

Maquette FUN-MOOC	Adresse MAC	Routeur par défaut	Routage IP
Client 192.168.1.1	00 :0c :29 :b8 :da :45	192.168.1.10	Désactivé
Attaquant 192.168.1.2	00 :0c :29 :b9 :6b :61	192.168.1.10	Activé
Passerelle 192.168.1.10 192.168.2.10	00 :0c :29 :4f :87 :63 00 :0c :29 :4f :87 :6d	Pas de routeur par défaut	Activé
Serveur 192.168.2.1	00 :0c :29 :4f :da :fa	192.168.2.10	Désactivé

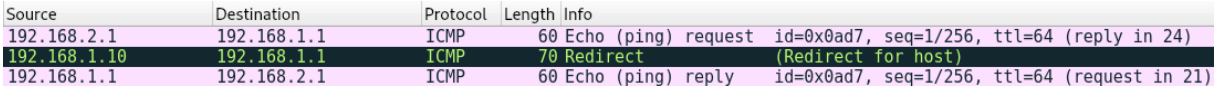
4.2. Attaque avec l'outil Ettercap

Lors de l'attaque avec l'outil Ettercap, on constate que le message ICMP redirect envoyé par l'attaquant est ignoré par la victime. En effet, une machine n'accepte les redirect seulement si le message provient de sa passerelle par défaut. Dans le cas ici, la source de ce paquet ICMP redirect n'est pas le routeur (le src n'est pas spoof donc la victime n'acceptera pas le message).

Les machines d'aujourd'hui ont des sécurités qui empêche toute modification de la table de routage. L'outil Ettercap est obsolète dans le cas d'une attaque MITM avec ICMP Redirect.

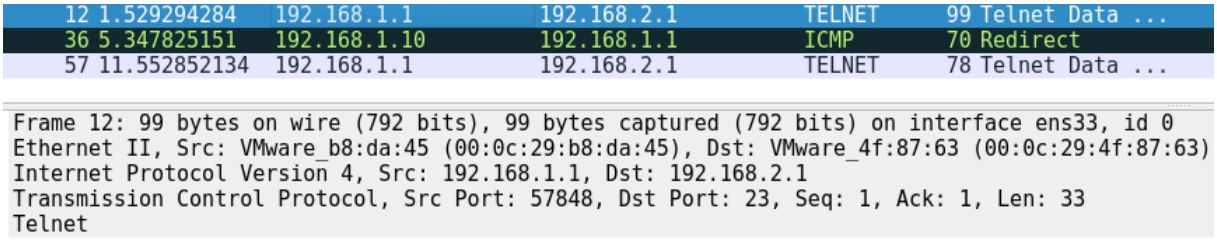
4.3. Attaque avec l'outil Redirect

Capture de trame sur l'attaquant après utilisation de l'outil redirect

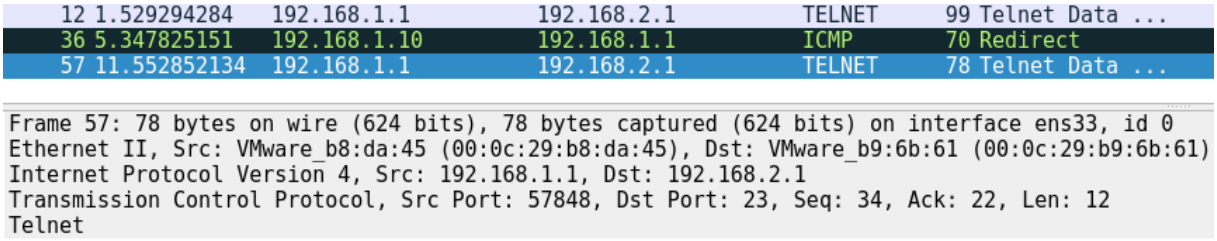


Source	Destination	Protocol	Length	Info
192.168.2.1	192.168.1.1	ICMP	60	Echo (ping) request id=0x0ad7, seq=1/256, ttl=64 (reply in 24)
192.168.1.10	192.168.1.1	ICMP	70	Redirect (Redirect for host)
192.168.1.1	192.168.2.1	ICMP	60	Echo (ping) reply id=0x0ad7, seq=1/256, ttl=64 (request in 21)

L'outil Redirect permet de déclencher un échange ICMP entre un serveur et une victime. Le serveur envoie un ping (echo request) à la victime, puis l'attaquant envoie l'ICMP Redirect en se faisant passer pour le routeur pour dire à la victime « si tu veux joindre 192.168.2.1 passe par 192.168.1.2 », puis la victime met alors à jour sa table de routage et poursuit la communication. Ces paquets ont pour but de simuler un échange ICMP entre la victime et le serveur en injectant discrètement la route de l'attaquant à la table de routage de la victime à l'aide de l'ICMP Redirect.



Comme on le voit dans la capture, avant le message redirect, le client (b8 :da :45) envoie comme prévu un paquet vers le routeur (4f :87 :63) qui est sa passerelle par défaut. Tout va bien avant l'attaque



Mais après le message on remarque que l'adresse MAC de destination a changé. En effet, 00 :0c :29 :b9 :6b :61 correspond à l'adresse MAC de l'attaquant. Cela signifie que la victime a accepté le message ICMP Redirect puisqu'il a mis à jour sa table de routage. L'attaque a fonctionné avec l'outil Redirect